On Extending ESnet's OSCARS with a Multi-Domain Anycast Service

Mark Boddie*, Timothy Entel*, Chin Guok†, Andrew Lake†, Jeremy Plante*, Eric Pouyoul†,
Bharath H. Ramaprasad*, Brian Tierney†, Joan Triay‡*, and Vinod M. Vokkarane*§
*Department of Computer and Information Science, University of Massachusetts, Dartmouth, MA, USA
†Lawrence Berkeley National Laboratory/ESnet, CA, USA
‡Department of Telematics Engineering, Universitat Politècnica de Catalunya (UPC), Castelldefels, Spain
§Claude E. Shannon Communication and Network Group, Massachusetts Institute of Technology (MIT), MA, USA
E-mail: vvokkarane@ieee.org

***Abstract*—Current scientific data applications require advanced network provisioning systems to support the transport of large volumes of data. Due to the availability of diverse computing and Grid clusters, these applications can benefit from anycasting capabilities. In contrast to unicasting, anycast routing allows the selection of a node from a group of candidate destinations. This new means of communication allows for greater routing flexibility and better network resource consumption. However, current provisioning systems do not provide fully compliant anycast implementations. In this paper, we extend ESnet's OSCARS virtual circuit provisioning system with anycast routing capabilities to support destination-agnostic applications on single- and multi-domain network scenarios. The proposed implementation significantly improves provisioning success over the native unicast implementation in compliance with the existing OSCARS framework.**

***Index Terms*—RWA, advance reservation, path computation, anycast, VLAN, PCE, OSCARS, Virtual Circuits.**

I. Introduction

A growing number of scientific computing applications require reliable services involving large amounts of data with varying quality of service (QoS) requirements. Furthermore, advanced multi-layer/multi-domain control planes are able to provide differentiated services demanded in next-generation networks. In the scope of this framework, both immediate reservation (IR) and advance reservation (AR) are needed to guarantee real-time service provisioning and network resource availability, respectively. In the former, resources are reserved and used immediately following the request. However, IR may be unable to guarantee the availability of resources with high probability. A more flexible resource provisioning can be accomplished through the implementation of advance reservation mechanisms [1]. These are of special interest for Grid applications. Such network services are supported by advanced on-demand network provisioning systems.

One such system is the On-Demand Secure Circuits and Advance Reservation System (OSCARS) [2] which is currently deployed on the Department of Energy's (DOE's) nation-wide Energy Sciences network (ESnet) [3]. The OSCARS software manages and automates the network functions based on user-specified requirements. OSCARS provides multi-domain, high-bandwidth virtual circuits that guarantee end-to-end network data transfer. These virtual circuits support scientific areas ranging from high energy physics applications, such as those performed on the Large Hadron Collider, to biological and environmental research. Such applications account for approximately fifty percent of the ESnet's gargantuan 60 petabytes of annual traffic [3].

Currently, OSCARS only allows for provisioning of unicast circuits as a service, i.e., a virtual circuit between a given source and a specified destination node (and port). However, there exist several destination-agnostic applications that can take advantage of anycast request provisioning, such as database replication and off-site backups. *Anycasting* refers to the transmission of data from a source node to any one member among a candidate destination set. A single anycast request r_A can be defined as a 5-tuple: $r_A = (s, D_s, \alpha, \tau, R)$. Specifically, a given source node s can select one of the m nodes in the destination set $D_s = \{d_1, d_2, \ldots, d_m\}$. When $|D_s| = 1$, the request is unicast. The variables α and τ denote the start and end time of r_A, and R is the desired transmission rate of a circuit satisfying the request. Carefully routing anycast requests can help carry additional traffic demands in the network [4]. Moreover, the anycast communication paradigm can help clients to find appropriate Grid resources, not only based on the actual available network resources, but also based on the Grid computing servers [5]. Anycasting also provides the ability for a request to access the resources efficiently in a network, thereby reducing energy consumption [6].

A key issue when dealing with multi-domain environments is to define what domain parameters (e.g., switching capability) should be disseminated among the domains, and whether the dissemination of such parameters is necessary at all. This is an important question to answer as network resource capacity may be wasted due to the mismatch between different domain's granularity. Solutions must be scalable and achieve a level of optimality, where the latter can be inferred as the traffic-engineering path chosen in the idealized case of a "flat" network, that is, no partitioning with a global state [7].

In this paper, we present a new multi-domain path computation element (PCE) implementation for the OSCARS framework that takes advantage of the anycast paradigm. The new anycast PCE modules will allow researchers to execute future destination-agnostic applications over ESnet, thus broadening the number of available services, and improving the network resource utilization globally. Furthermore, we extend the OSCARS framework to not only perform intra-domain path anycast computation, but also extend such computation across different network domains managed by different instances of

OSCARS for making inter-domain anycast path computation possible. Overall, in this paper we demonstrate the feasibility of our proposed implementation and evaluate the improvement of the anycast routing over unicast.

The remainder of this paper is organized as follows: Section II introduces the multi-domain capabilities of OSCARS. Section III of this paper describes the proposed anycast PCE implementation and Section IV describes the proposed multi-domain workflow of our anycast PCE. The performance comparisons with the baseline unicast implementation are given in Section V. Finally, Section VI concludes the paper.

II. OSCARS Multi-Domain Capabilities

OSCARS aims at exploring composable services and configuring highly modular, atomic network services on-demand via SOAP based web services. OSCARS is composed of several modules whose tasks include authorization, resource management, path computation, and inter-domain control management. Fig. 1 shows the relationship between the different OSCARS modules. The core of the OSCARS architecture is comprised of the Coordinator and the PCE framework. The Coordinator essentially handles the entire reservation workflow by managing the control flow between different modules. The PCE framework is responsible for identifying and computing the virtual circuit for a given AR, which is in-turn stored in the Resource Manager. The Resource Manager then spools for reservations which are due to be active in the next time interval and triggers the technology-specific path setup module to rig up the circuit for each such pending AR request.

OSCARS also supports the inherently multi-domain environment of large-scale science by allowing inter-operation with similar services in other network domains. In this context, OSCARS is also an IDC. DICE (Dante, Internet2, Canarie, ESnet) consortium [8], standardized the IDC protocols to set up end-to-end circuits across multiple domains with diverse circuit signaling protocols. These domains exchange topology information containing at the very least, the potential virtual circuit (VC) ingress and egress points. The VC setup request (via IDC protocol) is initiated at one end of the circuit and passed from domain to domain as the VC segments are authorized and reserved.

For inter-domain AR requests (requests spanning multiple OSCARS domains), OSCARS uses a non-RSVP-style signaling across domain boundaries to signal the circuit setup. The solution proposed by OSCARS to accomplish the multi-domain circuit setup is two-fold. Firstly, explicit agreement/registration between IDCs managing each domain are established so that they are mutually aware of each other's controllers, given the need to contact a particular domain as part of circuit setup for an inter-domain request. Secondly, the IDCs communicate using the OSCARS IDC protocol in accordance with local-domain policy along with available resources, which determines the setup of the inter-domain AR request. These inter-domain circuits are terminated at the domain boundary. Subsequently, a separate data plane

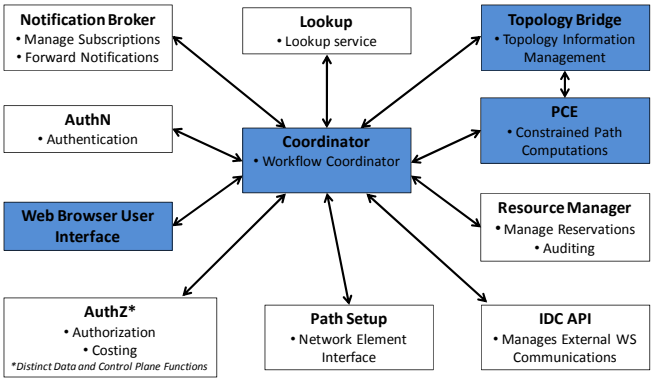


Fig. 1. OSCARS modular framework.

service is used to stitch the circuits together into an end-to-end path. In contrast to other approaches, data plane connection is facilitated by a helper process, not by signaling across domain boundaries. Each network domain provides its own control plane functions for circuit definition. Interestingly, the OSCARS IDC has successfully interoperated with several other IDCs to set up end-to-end path, cross-domain circuits.

III. Anycast Path Computation Element Implementation

The PCE is responsible for computing a single path given the existing network topology, and a connection request. In OSCARS 0.6, this service is provided as a framework which allows third-party PCE implementations to be developed and deployed alongside the rest of OSCARS' modules. The four main modules involved in the path computation request flow are the user interface (or IDC API), coordinator, topology bridge, and PCE modules (shaded modules in Fig. 1).

Like the rest of the OSCARS framework, PCEs are modules and each one is represented within the OSCARS Coordinator by a PCE Proxy that handles the communication between the Coordinator and the PCE. Requests to PCEs are assumed to be asynchronous. The PCE framework provides:

- **Modularity:** each PCE is executed as an independent process.
- **Distribution:** PCEs can be deployed on different (virtual or physical) hosts other than the OSCARS IDC host.
- **Security:** PCEs follow the OSCARS 0.6 security model in regard to authentication, authorization, and accounting.
- **Language neutrality:** while the default binding is JAVA, the APIs are based on web-services, thus allowing for independent developers to use any language as long as they comply with the API specification.

OSCARS 0.6 allows several PCEs to be deployed, each one of them responsible for computing a specific subset of local paths in a given domain. The execution process is defined as a flexible PCE workflow module, whereby purpose-specific component PCEs are connected in a workflow graph to incrementally prune network resources that do not meet the constraints of the user or network operator. As such, the output from one module can then be fed as input to the next.

Specifically, our proposed anycast PCE processes a network topology (domains + nodes + ports + links) as input and outputs a single path from the source to a selected destination.

Following the unicast model, our proposed anycast PCE is composed of four core modules which take as an anycast request and a network topology as input, and output an updated, pruned topology (refer to Fig. 2):

- **AnycastConnectivityPCE**: This PCE module is responsible for computing the network topology corresponding to the network connectivity graph between the source node and all the candidate destination nodes of the anycast group. The output of this module is an updated topology with node-pairs not physically connected by a physical fiber pruned out. This module is responsible for dynamically interpreting the network domain so that all other PCEs do not improperly assume additional connectivity.
- **AnycastBandwidthPCE**: This PCE removes the links, ports, and nodes that do not guarantee the bandwidth capacity of the user's anycast request. Fibers which are oversubscribed at the starting time of the request will be pruned from the topology. The behavior of this PCE is largely responsible for the existence of resource-driven connection blocking. In Section V we show how the probability that requests will be blocked is reduced as an effect of utilizing anycast communication.
- **AnycastVlanPCE**: Each port on a node has a designated number of VLAN tags which represents the maximum number of virtual circuits which may be accommodated at that node. The *AnycastVlanPCE* module prunes out the links, ports and nodes that do not have enough VLAN tags to support the virtual circuit., thereby guaranteeing secure connection establishment for all successfully provisioned requests.
- **AnycastDijkstraPCE**: This PCE module computes the potential end-to-end paths to each destination in the anycast set and then selects the final destination based upon some criteria. In this work, we select destinations to satisfy an anycast request such that the candidate along the shortest path is preferred. Alternative metrics can easily be incorporated with our existing *AnycastDijkstraPCE* design to select destinations based on a path's available bandwidth, and/or other metrics.

The worst-case runtime complexity of our anycast PCE implementation is increased over its unicast counterpart by a factor of $|D_s|$, the number of destinations in the anycast set.

OSCARS is responsible for providing the understanding of inter-module relationships and the ordering of the module executions. A *PCERuntime* agent controls this ordering through a customizable XML configuration file that prescribes rules for arranging the PCE module executions. PCE modules need not be aware of the relative execution ordering. The *NullAggregator* module aggregates a set of paths based on the result from several PCEs. In our case, the *NullAggregator* captures the result, *Tag 1* (refer (5) in Fig. 2), from the last

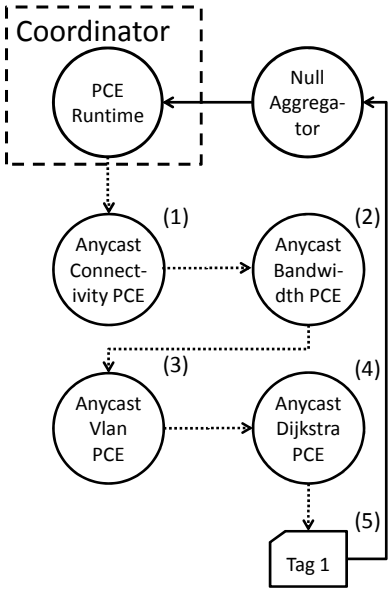


Fig. 2. Anycast PCE stack flow-chart.

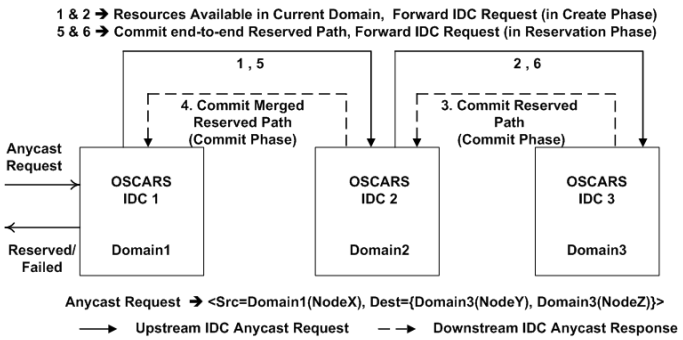


Fig. 3. Multi-domain anycast in OSCARS.

PCE to execute, *AnycastDijkstraPCE*. The final reply is sent back to the *PCERuntime* module, which governs the request forwarding between PCE modules. The final output from the execution of the anycast PCE workflow is a pruned topology consisting exclusively of the VC along the path from the source to the selected anycast destination.

IV. Multi-Domain Anycast AR Request Workflow

The multi-domain workflow for an anycast AR request is shown in Fig. 3. For the sake of simplicity, consider an IDC as

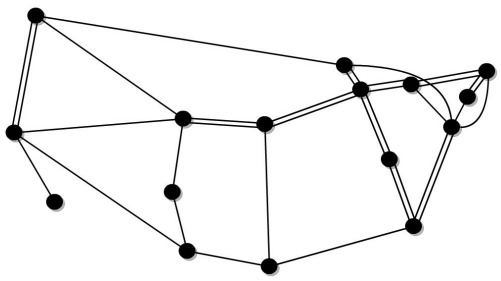


Fig. 4. 16-node ESnet SDN core network topology.

a single OSCARS instance. A multi-domain anycast request is first submitted to the local IDC (source IDC is IDC 1 in this case). In this example workflow, the anycast request specifies Node X in Domain 1 which is found locally in the network managed by IDC 1. The request also specifies Node Y and Node Z as part of the anycast destination set, which is remote to IDC 1. Now the Coordinator in IDC 1 initializes the PCE workflow. The *anycastConnectivityPCE* loads all the partially (ingress and egress only) or fully visible (sister network domains share entire topology) topologies as a topology stack to reach from the source to all the anycast destination domains. In this example Domain 2 and Domain 3 are loaded. The *anycastBandwidthPCE* and *anycastVlanPCE* then prune all the local nodes, ports, links in the topology stack which do not fit the user's constraints of bandwidth and VLAN. In case of MPLS, they simply prune the ingress and the egress nodes of the local domain. This pruned topology stack is then fed into the *anycastDijkstraPCE*, which finds the best local path to the egress node for all valid anycast destinations of the local domain and returns this path to the local Coordinator. Now the local Coordinator within IDC 1 determines that the request is inter-domain, flags the anycast request to be in the CREATE phase and forwards this request by loading the profile of the next inter-domain hop in the path which helps to communicate suitably over the inter-network with the next IDC responsible for the inter-domain hop. In Fig. 3, IDC 1 forwards the inter-domain anycast request to IDC 2. Now, IDC 2 performs actions similar to IDC 1 (the OSCARS coordinator and PCE framework are highly re-entrant and efficient by switching logic based on the phase a request is in). If a local path is found feasible, IDC 2 then forwards this request to IDC 3 which manages the destination domain, Domain 3. Now, IDC 3 performs actions similar to IDC 2 in computing the best path to all of the anycast destinations and returns whichever has the shortest number of hops back to the Coordinator. Upon successful receipt of the path, the Coordinator for IDC 3 then locally saves the path in its local database as reserved and changes the anycast request phase to COMMIT and forwards the request back to the sender of the request, IDC 2. IDC 2 sees the phase of the request to be COMMIT, and so it merges the local path with the global path and saves this merged path as the reserved path in the local database. IDC 2 again forwards the updated request to the original sender, IDC 1, which after merging the local and global paths, sets the full end-to-end path in its local database. Subsequently, IDC 1 changes the request status to RESERVED to indicate the end-to-end path is stitched and forwards this end to end reserved path to IDC 2. IDC 2 now overwrites the entire end-to-end anycast path again into its local database and forwards it to IDC 3 which performs similar action of persisting the end-to-end reserved path to the local database. IDC 3 flags the reservation as completed by setting the anycast request status to the RESERVATION-COMPLETE phase, which is then recursively transmitted back to IDC 1 (the original sender). The user is notified that the anycast AR request has been successfully reserved. If the request cannot be provisioned locally in the CREATE phase by any of the domains in the path, then the user is notified that the reservation has failed and the request is blocked.

V. Performance Results

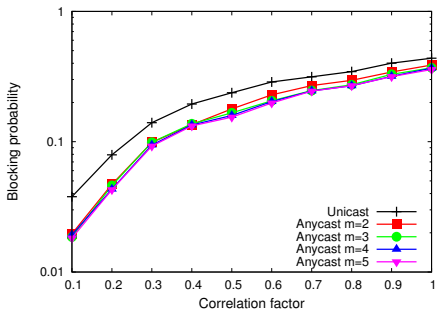
We have subjected our anycast PCE implementation to various dynamic traffic scenarios to examine the probability that requests will be blocked due to bandwidth and VLAN unavailability. Dynamic requests are those which arrive on a network, reserve network resources, and then depart from the network after some finite amount of time, thereby freeing up their resources for future requests. We consider various values of m for anycast m/1 connection requests and compare the associated average blocking probability and physical hop-count to that of the native unicast implementation. We measure the performance of our implementation on both single-domain networks and multi-domain networks.

A. Single-Domain Performance Results

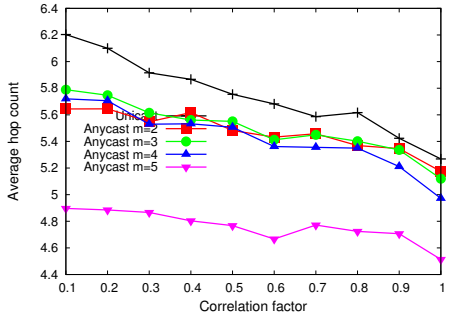
We simulate 30 unique sets of 100 AR requests (and present the average values in all scenarios) on the ESnet's science data network (SDN) core topology shown in Fig. 4. All links in the ESnet topology are bi-directional and are assumed to have 1 Gb/s bandwidth. The average nodal degree of ESnet is 4.06 and the average hop-distance of the topology is $5.02 hops$.[1] For each request, the source node and destination node(s) are uniformly distributed, while the request's bandwidth demands are uniformly distributed in the range [100 Mb/s, 500 Mb/s], in increments of 100 Mb/s. This allows for a realistic traffic scenario in which requests from different sites intend to transmit varied loads.

Fig. 5(a) plots the blocking probability against the AR request set's correlation factor. All requests are scheduled to reserve, transmit, and release network resources within two hours. This correlation factor corresponds to the probability that requests overlap during that two-hour window. The higher the correlation factor, the more requests overlap in time; a correlation factor of 1 corresponds to a set of dynamic immediate IR requests which arrive at time t = 0. The formula for calculating the correlation factor for a set of requests is given as $\sum_j C_j/n(n-1)$, where n is the number of requests to schedule, and C_j is the number of requests which overlap in time with request j [9]. Fig. 5(a) illustrates that the inclusion of our anycast PCE is sufficient enough to significantly lower blocking probability for both large and small correlation factors. Fig. 5(c) lists the percentage blocking improvement for each level of anycast over the existing unicast model across varied correlation factors (CF). In general, the higher the value of m, the greater the improvement over unicast. However, the expansion of the anycast destination set size provides little relatve benefit for $m > 2$. Blocking is reduced further as m increases, but the percentage improvement from m to $m + 1$ shrinks for larger m. This blocking reduction is only slightly advantageous when PCE execution times are

[1]OSCARS defines a hop as a port-to-port connection. Hops may be inter-nodal as well as intra-nodal.



(a) Blocking probability.

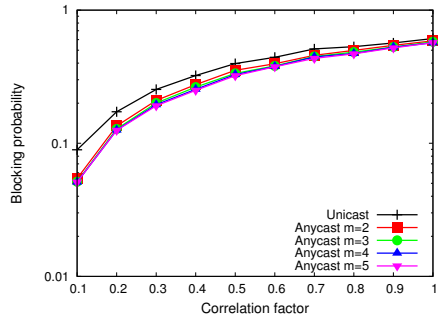


(b) Average path hop-count (inter-nodal + intra-nodal).

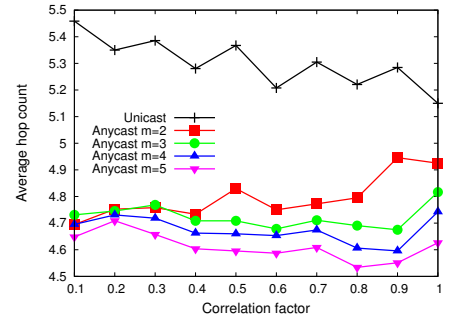
CF	m=2	m=3	m=4	m=5
0.1	48.11	50.94	49.06	51.89
0.3	29.59	29.34	32.65	33.93
0.5	24.81	29.92	33.08	35.04
0.7	14.30	21.57	22.02	22.25
0.9	14.44	17.91	20.77	21.21
1.0	11.43	15.76	16.16	18.20

(c) ESnet: % improvement over unicast.

Fig. 5. Performance comparison of our proposed anycast PCE for unicast and anycast requests m/1 on ESnet.



(a) Blocking probability.

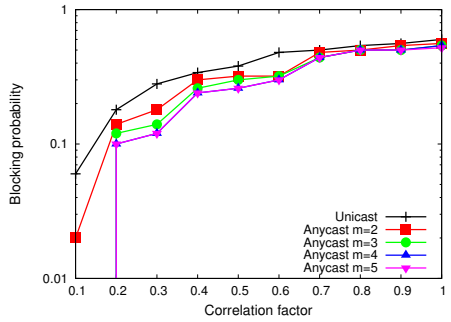


(b) Average path hop-count (inter-nodal + intra-nodal).

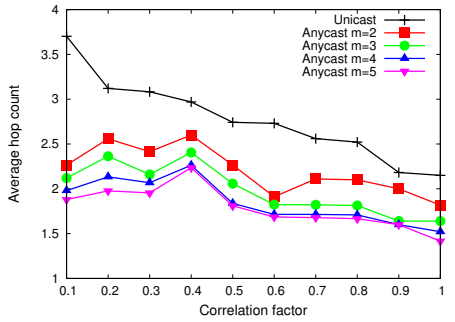
CF	m=2	m=3	m=4	m=5
0.1	39.18	42.54	42.91	42.91
0.3	17.35	21.02	23.26	24.70
0.5	11.00	15.70	17.38	18.98
0.7	10.44	12.33	12.92	15.07
0.9	4.056	6.82	7.11	8.23
1.0	3.924	5.50	7.08	6.87

(c) GÉANT: % improvement over unicast.

Fig. 6. Performance comparison of our proposed anycast PCE for unicast and anycast requests m/1 on GÉANT.



(a) Blocking probability.



(b) Average path hop-count (inter-nodal).

CF	m=2	m=3	m=4	m=5
0.1	66.67	100.0	100.0	100.0
0.3	35.71	50.00	57.14	57.14
0.5	15.79	21.05	31.58	31.58
0.7	4.00	12.00	12.00	12.00
0.9	3.57	10.71	10.71	10.71
1.0	6.67	10.00	10.00	13.33

(c) Multi-domain: % improvement over unicast.

Fig. 7. Performance comparison of our proposed anycast PCE for multi-domain unicast and anycast requests m/1 on ESnet - GÉANT.

considered. As the request set size grows, so does the number of loop iterations in each anycast PCE module.

Fig. 5(b) captures the average physical hop-count of successfully provisioned requests; anycast is able to reduce the average number of hops compared to unicast. As the size of the destination set increases, the number of available destinations also increases. Since our *AnycastDijkstraPCE* module routes to the destination along the shortest path (if bandwidth restrictions permit), the likelihood of successfully routing to a nearby neighbor increases. For this reason, as m grows, the average number of hops of satisfied requests shrinks, without adversely affecting the blocking probability. Anycast scenarios corresponding to all values of m (including m = 1) display a similar downward trend in the number of physical hops as the correlation factor increases. This trend is due to the corresponding blocking probabilities shown in Fig. 5(a). As large quantities of traffic enter the network, the bandwidth reserved at any time will be quite high. This of course leads to longer VCs (those needing more resources to successfully route) being blocked. Only the shortest paths will be successful at these high loads. Fig. 5 showcase the ability for anycast communication to efficiently manage resource utilization on the ESnet.

We have also evaluated our anycast PCE implementation on an augmented 13-node version of the Gigabit European Advanced Network Technology (GÉANT) topology used throughout Europe for research and education [10]. This version of GÉANT is connected to ESnet through an inter-

domain link and has an average hop-distance of 4.84 and an average nodal degree of 2.77. Our simulations for this topology are structured similarly to ESnet, such that all links are considered bi-directional with uniform bandwidth of 10 Gb/s, while requests fill the range [1 Gb/s, 5 Gb/s] in increments of 1 Gb/s. The low nodal degree will force more paths to occupy the same links, which increases contention for circuit establishment. Fig. 6(a) shows how this high contention can increase the blocking probability. In general, the same trend is observable as that shown for ESnet, but the blocking is higher for all values of m in GÉANT. Further, Fig. 6(c) displays the percentage blocking reduction over unicast. The savings are noticeably lower than on ESnet, however, it can be similarly observed that larger values of m provide lower blocking. Fig. 6(b) shows the average hop-count for established paths on GÉANT. The average size of unicast paths is nearly one hop longer than anycast path sizes. Due to the high contention caused by the low nodal-degree of the network, as correlation grows, more short paths will monopolize the same links, thus allowing only longer paths to be used to satisfy anycast requests. This explains the spike in hop-count for correlation values > 0.9.

B. Multi-Domain Performance Results

In a multi-domain scenario, each IDC in the anycast PCE performs pruning of bandwidth and VLANs locally in its own domain. This pruning takes place in all domains which are relevant in reaching the destination(s). Here we consider two metrics, blocking probability, and average $inter-nodal$ hop-count for 5 unique sets of AR requests, each consisting of 50 multi-domain AR requests. We further assume that the source of these requests is always in the ESnet domain and the destination(s) always lie(s) in the GÉANT domain. The ESnet and GÉANT domains are connected by two inter-domain links, each of capacity 10 Gb/s. We also assume that every link in both domains have a maximum reservable capacity of 10 Gb/s. The AR request sizes vary in bandwidth uniformly within the range [1 Gb/s, 5 Gb/s] with a granularity of 1 Gb/s.

As shown in Fig. 7(a), the blocking probability for requests in the multi-domain setup are considerably high for unicast at low and medium correlations as compared to that of anycast with $m \geq 2$. Anycast allows us to achieve significantly better blocking performance (about 33% better on average) even at anycast cardinality of m=2, as shown in Fig. 7(c). As the anycast cardinality increases, chances of finding an alternative path increases, which is reflected upon by the significant improvement in blocking performance up to a maximum of 42.85% increase when compared with the traditional unicast. We can observe that at higher correlation factor values ≥ 0.7, anycast performs slightly better (about $8-10\%$ better on average) than unicast in terms of blocking performance. This significant blocking reduction is backed by smaller average hop-count for anycast requests, as shown in Fig. 7(b). We can observe from the figure that anycast offers lowered average hop-count by about 1 hop on average, even at anycast m=2, and the best average hop-count at anycast m=5, reducing the hop-count on a average by about 1.45 hops over unicast. Thus by introducing the proposed anycast PCEs for a multi-domain setup, we observe that it greatly reduces the blocking of AR requests and reduces the overall signaling in the network by reducing the hop-count significantly.

VI. Conclusion

In this paper we have described the proposed implementation of an anycast advance reservation virtual circuit reservation service to the existing OSCARS framework for use on the DOE's ESnet and the GÉANT topologies. By subjecting the existing unicast, and the new anycast PCE modules and configurations to dynamic traffic requests, we have shown that anycast provisioning significantly reduces the likelihood of request blocking, and is able to successfully provision AR requests while adhering to the existing OSCARS framework and architecture. Our results establish the cornerstone for a deployable AR connection establishment service for use on real-world core networks in an effort to provide anycast functionality for use in facilitating large-scale science applications. This addition of anycast circuit provisioning opens OSCARS to a vast new paradigm of destination-agnostic scientific areas beyond the unicast-only applications currently supported by ESnet and GÉANT.

Acknowledgment

This work has been partially supported by the Department of Energy COMMON project (grant DE-SC0004909). Joan Triay was a visiting researcher at University of Massachusetts supported by a Fulbright graduate fellowship.

References

[1] Y.-D. Lin, C.-H. Chang, and Y.-C. Hsu, "Bandwidth brokers of instantaneous and book-ahead requests for differentiated services networks," *IEICE Trans. on Communications*, vol. 85, no. 1, pp. 278–283, 2002.

[2] C. P. Guok, D. W. Robertson, E. Chaniotakis, M. R. Thompson, W. Johnston, and B. Tierney, "A user driven dynamic circuit network implementation," in *Proc. Distributed Autonomous Network Management Systems Workshop (DANMS)*, New Orleans, LA, USA, Nov. 2008, pp. 1–5.

[3] Lawrence Berkeley National Laboratory, "Energy Sciences Network (ESnet)," [Online]. Available: http://www.es.net.

[4] D. Din, "A hybrid method for solving ARWA problem on WDM network," *Computer Communications*, vol. 30, no. 2, pp. 385–395, Jan. 2007.

[5] T. Stevens, M. D. Leenheer, C. Develder, F. Turck, B. Dhoedt, and P. Demeester, "Anycast routing algorithms for effective job scheduling in optical grids," in *Proc. of European Conference on Optical Communication (ECOC) 2006*, Cannes, France, Sep. 2006, pp. 371–372.

[6] B. G. Bathula, M. Alresheedi, and J. Elmirghani, "Energy efficient architectures for optical networks," in *Proc. London Communications Symposium (LCS) 2009*, London, UK, Sep. 2009.

[7] F. Aslam, Z. Uzmi, and A. Farrel, "Interdomain path computation: Challenges and Solutions for Label Switched Networks," *IEEE Commun. Mag.*, vol. 45, no. 10, pp. 94–101, Oct. 2007.

[8] DICE Consortium, "DICE," [Online]. Available: http://www.controlplane.net/.

[9] N. Charbonneau and V. M. Vokkarane, "Static routing and wavelength assignment for multicast advance reservation in all-optical wavelength-routed wdm networks," *IEEE/ACM Transactions on Networking*, 2011.

[10] Delivery of Advanced Network Technology to Europe (DANTE), "GÉANT," [Online]. Available: http://www.dante.net.